2

NPS-53-89-008

NAVAL POSTGRADUATE SCHOOL

Monterey, California

NUMERICAL EXPERIENCE WITH A SUPERFAST
REAL TOEPLITZ SOLVER

William Gragg
Gregory S. Ammar

February 1989

Approved for public release; distribution unlimited

Prepared for: Naval Postgraduate School and the
National Science Foundation, Washington
D.C. 20550

NAVAL POSTGRADUATE SCHOOL
Department of Mathematics

This report was prepared in conjunction with research conducted for the National Science Foundation and for the Naval Postgraduate School Research Council and funded by the Naval Postgraduate School Research Council.

Prepared by:

William Gragg
WILLIAM GRAGG
Professor of Mathematics

Reviewed by:

HAROLD M. FREDRICKSEN
Chairman
Department of Mathematics

Released by:

Kneale T. Marshall
KNEALE T. MARSHALL
Dean of Information
and Policy Sciences

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): NPS-53-89-008	5 MONITORING ORGANIZATION REPORT NUMBER(S): NPS-53-89-008
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable): 53	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School and the National Science Foundation
6c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943	7b ADDRESS (City, State, and ZIP Code): Monterey, CA 93943 and Washington, D.C. 20550
8a NAME OF FUNDING/SPONSORING ORGANIZATION: Naval Postgraduate School; 8b OFFICE SYMBOL (If applicable): 53	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: O&MN, Direct funding
8c ADDRESS (City, State, and ZIP Code): Monterey, CA 93943	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO / PROJECT NO / TASK NO. / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

NUMERICAL EXPERIENCE WITH A SUPERFAST REAL TOEPLITZ SOLVER

12 PERSONAL AUTHOR(S)

William Gragg and Gregory S. Ammar

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Technical Report	FROM 1 Oct 88 TO 30 Jan 89	1989 February 22	

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	fast Schur algorithm, fast Toeplitz solver, numerical stability

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

We briefly describe the Generalized Schur Algorithm for the superfast solution of positive definite Toeplitz systems of equations and its relationship with Schur's algorithm and the Szegö recursions. We then present some experimental results obtained with our FORTRAN implementation of this superfast Toeplitz solver. We will see that the algorithm displays favorable behavior in that the growth rates of the resulting residuals and errors are comparable with those of the Szegö recursions.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL: William Gragg	22b TELEPHONE (Include Area Code): (408) 646-2194	22c OFFICE SYMBOL: 53Gr

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

NUMERICAL EXPERIENCE WITH A SUPERFAST REAL TOEPLITZ SOLVER†

Gregory S. Ammar
Department of Mathematical Sciences
Northern Illinois University
DeKalb, IL 60115

William B. Gragg
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

Abstract

We briefly describe the Generalized Schur Algorithm for the superfast solution of positive definite Toeplitz systems of equations and its relationship with Schur's algorithm and the Szegö recursions. We then present some experimental results obtained with our FORTRAN implementation of this superfast Toeplitz solver. We will see that the algorithm displays favorable behavior in that the growth rates of the resulting residuals and errors are comparable with those of the Szegö recursions.

1. Introduction

The problem of efficiently solving a linear system of equations $M\,x = b$ where M is an Hermitian positive definite Toeplitz matrix is of considerable interest in many areas of mathematics and engineering. While standard Gauss or Choleski methods for solving such a system of order n require $O(n^3)$ arithmetic operations, there are several algorithms for obtaining the solution in $O(n^2)$ operations [21,24,5,18,13]. We refer to these algorithms as *fast Toeplitz solvers*. More recently, algorithms have been presented for solving a Toeplitz system of equations using $O(n\log^2 n)$ arithmetic operations [11,6,7,15,17,22]. These *superfast*

† Presented at the International Conference on Linear Algebra and Its Applications, Valencia, Spain, September, 1987. This research was supported in part by the National Science Foundation under grant DMS-8704196.

Toeplitz solvers are all based on certain doubling procedures. While they are more complicated to implement than the fast methods, they require substantially less computational work for sufficiently large problems.

Current fast and superfast Toeplitz solvers are numerically unstable, and therefore unreliable, when applied to arbitrary Toeplitz systems [8]. However, Cybenko [9, 10] has shown that the algorithms of Levinson, Durbin, and Trench are numerically stable for the class of positive definite Toeplitz matrices in the sense that accurate results can be expected when the problem is well conditioned. Stability of some superfast algorithms can also be expected for the class of positive definite matrices [8]. We remark that some fast and superfast algorithms are also potentially forward stable for other classes of matrices, for example, for totally positive Toeplitz matrices.

The numerical reliability of fast and superfast Toeplitz solvers is a topic of ongoing research. Since they are relatively new on the scene and, moreover, implementations of these algorithms are less tractable than those of the fast algorithms, numerical results for superfast Toeplitz solvers are currently rare. Knowledge of the numerical reliability is important because superfast methods allow for the solution of problems which would require a prohibitive amount of time for a fast method. We are not aware of any experimental results for any superfast Toeplitz solvers except for those presented in [2].

Positive definite Toeplitz matrices play an important role in several aspects of classical analysis including the classical theory of Szegö polynomials, the trigonometric moment problem, and the Carathéodory coefficient problem [1, 16]. Several Toeplitz solvers are direct manifestations of algorithms and formulas that were developed in the classical theory (see, e.g., [20, 19, 3]). In [3], the superfast Toeplitz solver presented in [17] and [22] is described in terms of a generalization of Schur's classical algorithm.

In this paper we give a general description of the Generalized Schur Algorithm and its use in the superfast solution of Toeplitz systems. We then present

some results of our experimental code for the superfast solution of real positive definite Toeplitz matrices. Refer to [3] for a development of the Generalized Schur Algorithm and its relation with fast Toeplitz solvers, and see [2,4] for descriptions of the implementation of the algorithm for complex and real positive definite Toeplitz matrices.

2. Some Toeplitz Solvers

The Toeplitz solvers we consider can be divided into two phases: a factorization phase and a solution phase. In this section we summarize some fast Toeplitz solvers based on the Szegö recursions and Schur's algorithm. We then present the generalization of Schur's algorithm for the superfast solution of a positive definite Toeplitz system of equations.

2.1. The Szegö Recursions. The Hermitian positive definite Toeplitz matrix $M = M_{n+1} = [\mu_{j-k}]_{j,k=0}^{n} = M^*$ defines an inner product on the set of polynomials of degree at most n by

$$<\lambda^j, \lambda^k> := \mu_{j-k}$$

The monic polynomials χ_j, $\deg \chi_j = j$, $j = 0, \cdots, n$ that are orthogonal with respect to this inner product are the *(monic) Szegö polynomials* determined by M. These polynomials can be constructed using roughly n^2 multiplications and additions by the Szegö recurrence relations.

Szegö Recursions:

$$\begin{array}{l}
\chi_0 := 1, \quad \delta_0 := \mu_0, \\
\textbf{for } j = 0, 1, \ldots, n-1 \\
\quad \gamma_{j+1} := - < 1, \lambda \chi_j > / \, \delta_j \\
\quad \chi_{j+1}(\lambda) := \lambda \chi_j(\lambda) + \gamma_{j+1} \tilde{\chi}_j(\lambda) \\
\quad \delta_{j+1} := \delta_j \, (1 - |\gamma_{j+1}|^2)
\end{array}$$

where $\tilde{\chi}_j(\lambda) := \lambda^j \, \overline{\chi}(1/\lambda)$ is the polynomial obtained by conjugating and reversing the coefficients of χ_j. The quantities γ_j are called *Schur parameters*; they serve as intermediate quantities in the Szegö recursions and are often of physical and mathematical significance.

The Szegö recursions comprise the first phase of several Toeplitz solvers. In fact, they are equivalent with the Levinson-Durbin for the solution of the Yule-Walker equations of linear prediction. Define R to be the unit right triangular matrix whose kth column contains the coefficients of χ_k $(0 \leq k \leq n)$. By definition of the Szegö polynomials we have

$$R^* M R = D \, , \quad M^{-1} = R \; D^{-1} R^{-*},$$

where $D := \mathrm{diag}[\delta_k]_0^n$ and $\delta_k := <\chi_k, \chi_k>$. Thus, the Szegö polynomials determine the *reverse Choleski factorization* of M^{-1}. This is the factorization that underlies the solution phase of Levinson's algorithm for the solution of a Toeplitz system of equations.

In Trench's algorithm a relationship between the entries of M^{-1} is used to construct M^{-1} from its last column, which is given by χ_n and δ_n. This relationship follows from the Christoffel-Darboux formula for Szegö polynomials [24, 20]. The matrix interpretation of the Christoffel-Darboux-Szegö formula is the *Gohberg-Semencul formula*

$$\delta_n \, M_{n+1}^{-1} = T_1 \, T_1^* - T_0^* \, T_0$$

where

$$\begin{bmatrix} T_0 \\ T_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \rho_0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \rho_{n-1} & \cdots & \rho_0 & 0 \\ 1 & \rho_{n-1} & \cdots & \rho_0 \\ 0 & 1 & \ddots & \cdots \\ \vdots & \ddots & \ddots & \rho_{n-1} \\ 0 & \cdots & 0 & 1 \end{bmatrix},$$

and $\chi_n(\lambda) = \sum_0^n \rho_j \lambda^j$, $(\rho_n \equiv 1)$. See, e.g., [12]. Thus, the Gohberg-Semencul formula provides a factorization of M^{-1} that is determined by the last Szegö polynomial χ_n and its squared norm δ_n. The product $M^{-1}b$ can then be obtained in $O(n \log n)$ operations by using Fast Fourier Transform (FFT) techniques. This provides an $O(n \log n)$ procedure for the second phase of a Toeplitz solver.

2.2. Schur's Algorithm. Another class of Toeplitz solvers is based on the classical algorithm of Schur [23]. Let ϕ be a *Schur function*, which is a holomorphic mapping of the open unit disk in the complex plane into its closure. Schur's algorithm generates a continued fraction representation of a given Schur function $\phi = \phi_0$ by the successive application of linear fractional transformations (LFTs). Given a Schur function ϕ_{n-1}, define $\gamma_n = \phi_{n-1}(0)$ and

$$\phi_n(\lambda) := \frac{1}{\lambda} \, \frac{\phi_{n-1}(\lambda) - \gamma_n}{1 - \bar{\gamma}_n \, \phi_{n-1}(\lambda)}.$$

It is easy to see that ϕ_n is also a Schur function. Furthermore, if $|\gamma_n| = 1$ then $\phi_n(\lambda) \equiv \gamma_n$; in this case ϕ is a rational function and the algorithm terminates. Otherwise, we repeat the procedure to obtain ϕ_{n+1} from ϕ_n. In this way we obtain the *Schur continued fraction representation* of ϕ,

$$\phi = T_n(\phi_n) = t_1 \circ t_2 \circ \cdots \circ t_n(\phi_n),$$

where

$$t_n(\tau) = \frac{\gamma_n + \lambda \tau}{1 + \bar{\gamma}_n \lambda \tau}$$

The function $T_n(0)$ is referred to as the *nth approximant* of ϕ, and ϕ_n is called the *nth tail* of ϕ. Thus, Schur's algorithm effects a sequence of elementary linear fractional transformations on the initial Schur function ϕ, and the *Schur parameters* γ_j are intermediate quantities that determine the elementary linear fractional transformations.

To implement Schur's algorithm we write each Schur function as the quotient of formal power series, $\phi_n = \dfrac{\alpha_n}{\beta_n} = \dfrac{\sum_{j=0}^{\infty}\alpha_{n,j}}{\sum_{j=0}^{\infty}\beta_{n,j}}$, with $\beta_{n,0} > 0$ as a partial normalization. Organizing Schur's algorithm so that the coefficient pairs $\alpha_{0,k}, \beta_{0,k}$ are entered and *processed sequentially* leads to the progressive form of Schur algorithm, which is given below.

Schur's Algorithm:

for $k = 1, 2, 3, \cdots$ **while** $|\gamma_k| < 1$

  enter $\alpha_{0,k-1}, \beta_{0,k-1}$

  for $j = 1, 2, 3, \cdots, k-1$

$$\begin{bmatrix} \alpha_{j,k-j-1} \\ \beta_{j,k-j} \end{bmatrix} = \begin{bmatrix} 1 & -\gamma_j \\ -\bar{\gamma}_j & 1 \end{bmatrix} \begin{bmatrix} \alpha_{j-1,k-j} \\ \beta_{j-1,k-j} \end{bmatrix}$$

  $\gamma_k = \alpha_{k-1,0}/\beta_{k-1,0},$

  $\beta_{k,0} = \beta_{k-1,0}\,(1 - |\gamma_k|^2)$

If the polynomials $\alpha_0^{(n)}, \beta_0^{(n)}$ of degree less than n formed by the first n terms of α_0, β_0 are input, then the progressive Schur algorithm calculates $\alpha_k^{(n-k)}, \beta_k^{(n-k)}$

and γ_k, $k = 1, \cdots, n$ using roughly n^2 multiplications and additions.

In contrast with the Szegö recursions, Schur's algorithm provides a class of fast Toeplitz solvers based on the Choleski factorization of M, $M = L\ D\ L^*$, with L unit lower triangular (see, for example, [19, 22, 3]). In particular, we have the following.

Proposition 2.1. If $M = [\mu_{j-k}]_{j,k=0}^{n} = M^*$ is positive definite, then $\alpha_{0,j} = -\overline{\mu}_{j+1}$, $\beta_{0,j} = \overline{\mu}_j$, $j = 0,1, \ldots, n-1$, are the first n coefficients of formal power series α_0,β_0 such that $\phi_0 = \alpha_0/\beta_0$ is a Schur function.

Proposition 2.2. Let M, $\alpha_0^{(n)}$, $\beta_0^{(n)}$ be as in Proposition 2.1. Then

1. The Schur parameters obtained by Schur's algorithm applied to $\alpha_0^{(j)}$, $\beta_0^{(j)}$ are identical with the Schur parameters generated by the Szegö recursions applied to M.
2. The lower triangular matrix $L\ D = [\tau_{j,k}]_{j,k=0}^{n}$ is given by $\tau_{j,k} = \overline{\beta}_{k,j-k}$ $(k \leq j)$.

We remark that the connection between Schur functions and Toeplitz matrices is provided by the solution of the Carathéodory coefficient problem and the correspondence between Carathéodory functions and Schur functions. See, for example, [16,1]. Proposition 2.2 follows from by comparison of the resulting Szegö and Schur recursions.

Schur's algorithm can also be formulated in terms of the LFTs T_n. It is shown in [2, 3] that

$$T_n(\tau) = \frac{\xi_n + \tilde{\eta}_n \tau}{\eta_n + \tilde{\xi}_n \tau},$$

where ξ_n, η_n are polynomials that satisfy the recurrence relations

$$\begin{bmatrix} \xi_n \\ \eta_n \end{bmatrix} = \begin{bmatrix} \tilde{\eta}_{n-1} & \xi_{n-1} \\ \tilde{\xi}_{n-1} & \eta_{n-1} \end{bmatrix} \begin{bmatrix} \gamma_n \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \xi_0 \\ \eta_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.1}$$

and $\tilde{\xi}_n(\lambda) := \lambda^n\, \overline{\xi}_n(1/\lambda)$, $\tilde{\eta}_n(\lambda) := \lambda^n\, \overline{\eta}_n(1/\lambda)$. We refer to ξ_n and η_n, which

have degree less than n for $n > 1$, as the nth *Schur polynomials* associated with the Schur function ϕ_0. The Schur polynomials are generalizations of the Schur parameters in the sense that they determine compositions of the elementary linear fractional transformations that the Schur parameters determine.

The Schur polynomials are also important in that they can be used to obtain the Szegö polynomials.

Proposition 2.3. Let M, $\alpha_0^{(n)}$, $\beta_0^{(n)}$ be as in Proposition 2.1, and let ξ_k, η_k be the kth Schur polynomials determined by $\alpha_0^{(n)}$, $\beta_0^{(n)}$. Then the monic Szegö polynomials corresponding with M are determined by

$$\bar{\chi}_k(\lambda) = \tilde{\eta}_k + \tilde{\xi}_k / \lambda.$$

2.3. The Generalized Schur Algorithm. Let

$$T_{n,k} = t_{n+1} \circ t_{n+2} \circ \cdots \circ t_{n+k}$$

be the LFT generated by k steps of Schur's algorithm applied to ϕ_n, and let $\xi_{n,k}, \eta_{n,k}$ be the Schur polynomials corresponding with $T_{n,k}$. Note that for any $k > 0$ we have $\phi_n = T_{n,k}(\phi_{n+k})$; that is, the kth tail of ϕ_n is the $(n+k)$th tail of ϕ.

The Generalized Schur Algorithm is based on a doubling procedure for generating T_{2k} from T_k. It can be described in general terms as follows.

Generalized Schur Algorithm.

To obtain ξ_{2n}, η_{2n} from $\alpha_0^{(2n)}$, $\beta_0^{(2n)}$:

0. Obtain ξ_n, η_n from $\alpha_0^{(n)}$, $\beta_0^{(n)}$,
1. Obtain $\alpha_n^{(n)}$, $\beta_n^{(n)}$ from $\alpha_0^{(2n)}$, $\beta_0^{(2n)}$, ξ_n, η_n using $\phi_n = T^{-1}(\phi_0)$,
2. Obtain $\xi_{n,n}, \eta_{n,n}$ from $\alpha_n^{(n)}$, $\beta_n^{(n)}$ as ξ_n, η_n were obtained from $\alpha_0^{(n)}$, $\beta_0^{(n)}$,
3. Obtain ξ_{2n}, η_{2n} from ξ_n, η_n, $\xi_{n,n}, \eta_{n,n}$ using $T_{2n} = T_{0,n} \circ T_{n,n}$.

The algorithm is started by performing Step 0 directly by, for example, setting $\xi_1 = \alpha_{0,0}/\beta_{0,0}$ and $\eta_1 = 1$. More generally, we can use Schur's algorithm to generate $[\gamma_j]_1^{n_0}$ for a small value of n_0, and obtain ξ_{n_0}, η_{n_0} from the Schur parameters and the recursions (2.1). Now suppose Step 0 has been completed for some $n \geq n_0$.

We have

$$\phi_n = \frac{\alpha_n}{\beta_n} = T_n^{-1}(\phi_0) = \frac{\alpha_0\eta_n - \beta_0\xi_n}{\beta_0\tilde{\eta}_n - \alpha_0\tilde{\xi}_n}.$$

and in [2,3] it is shown that

$$\alpha_0\eta_n - \beta_0\xi_n = \gamma_{n+1}\,\delta_n\,\lambda^n + O(\lambda^{n+1}),$$

$$\beta_0\tilde{\eta}_n - \alpha_0\tilde{\xi}_n = \delta_n\lambda^n + O(\lambda^{n+1}).$$

We can therefore take

$$\alpha_n = (\alpha_0\eta_n - \beta_0\xi_n)/\,\lambda^n\ ,\ \ \beta_n = (\beta_0\tilde{\eta}_n - \alpha_0\tilde{\xi}_n)/\,\lambda^n$$

These equations allow us to obtain $\alpha_n^{(n)}$, $\beta_n^{(n)}$ from $\alpha_0^{(2n)}$, $\beta_0^{(2n)}$, ξ_n, η_n in Step 1.

Step 2 is the doubling step. We obtain $\xi_{n,n}$ and $\eta_{n,n}$ from $\alpha_n^{(n)}$ and $\beta_n^{(n)}$ as $\xi_{0,n} = \xi_n$ and $\eta_{0,n} = \eta_n$ were obtained from $\alpha_0^{(n)}$ and $\beta_0^{(n)}$.

Step 3, the composition of T_n and $T_{n,n}$, is given by

$$\xi_{0,2n} = \tilde{\eta}_{0,n}\,\xi_{n,n} + \xi_{0,n}\,\eta_{n,n},$$

$$\eta_{0,2n} = \tilde{\xi}_{0,n}\,\xi_{n,n} + \eta_{0,n}\,\eta_{n,n}.$$

Thus, the Generalized Schur Algorithm consists of various polynomial multiplications and additions performed in a recursive manner. The multiplication of polynomials can be efficiently performed using standard FFT techniques. This results in an $O(n\log^2 n)$ algorithm for computing ξ_n and η_n, where $n = 2^\nu$. The Toeplitz system of equations $M_{n+1}x = b$ can then be solved by forming χ_n from ξ_n, η_n by Proposition 2.3, and using the Gohberg-Semencul formula to perform the solution phase of the algorithm in $O(n\log n)$ arithmetic operations.

The recursive nature of the algorithm is easily handled in a programming language that allows self-referencing subroutines (e.g., PASCAL or APL). In such an environment it is easy to obtain a rudimentary implementation of the algorithm. However, in order to obtain a FORTRAN implementation of the algorithm we must consider the doubling procedure and resulting storage requirements more carefully.

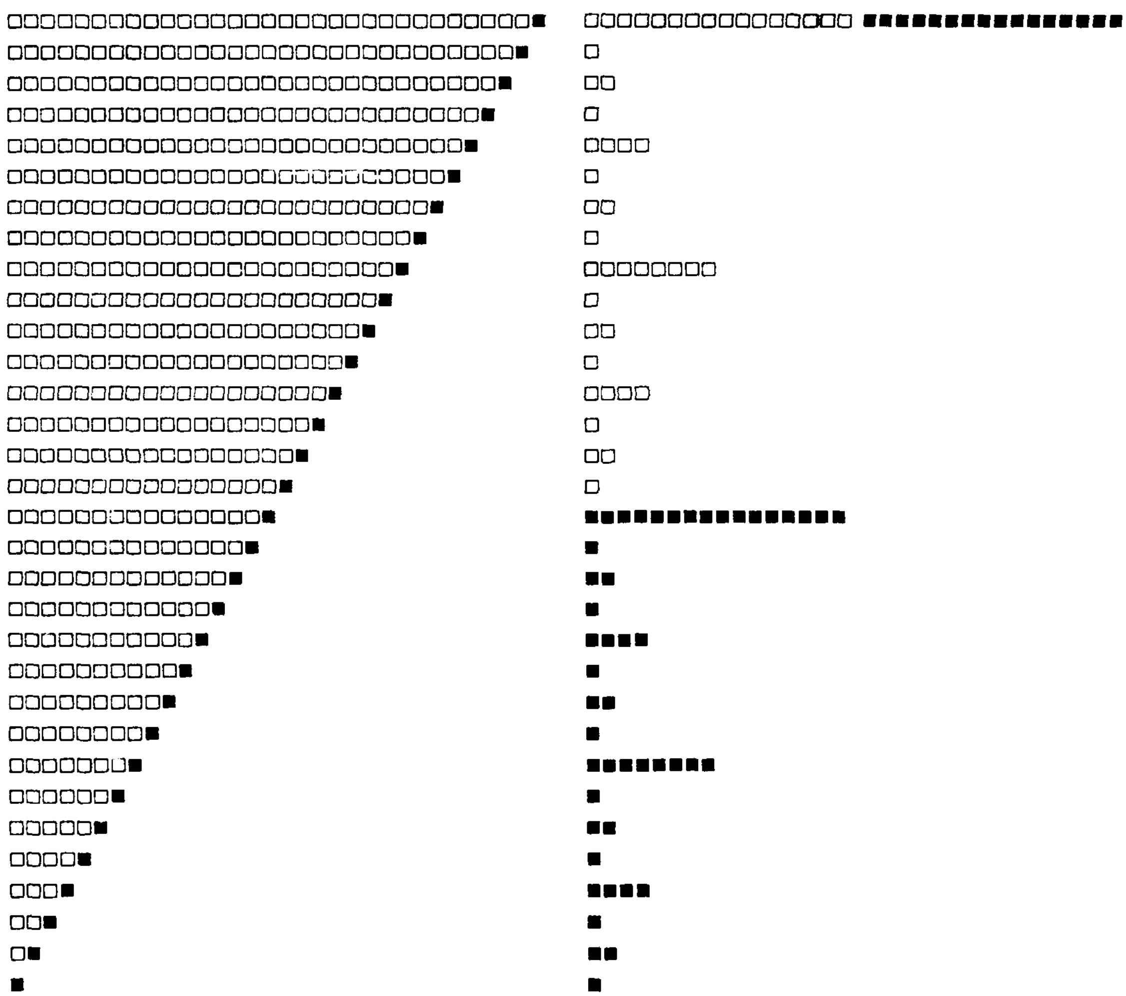
To contrast the Generalized Schur Algorithm with Schur's algorithm, the coefficients of α_k, β_k that are calculated by each algorithm are displayed in Figure 1 (for $n_0 = 1$). While the classical Schur algorithm generates $\alpha_k^{(n-k)}$, $\beta_k^{(n-k)}$, γ_k, $0 < k < n$ from $\alpha_0^{(n)}$, $\beta_0^{(n)}$, the number of coefficients of α_k, β_k calculated by the Generalized Schur Algorithm is equal to the largest power of two that divides k, unless this power of two is less than n_0. Thus, the Generalized Schur Algorithm computes only pieces of the Choleski factorization of the Toeplitz matrix. Nevertheless, *we can obtain every Schur parameter;* these parameters are often of physical and mathematical significance.

FIGURE 1

The kth row of squares represents the coefficients of α_k and β_k that are calculated by Schur's algorithm and the Generalized Schur Algorithm ($k = 0,\ldots,n-1$). The filled squares represent the quantities that are calculated during one iteration of each algorithm. In Schur's algorithm the coefficients of α_0, β_0 are processed one at a time. In the Generalized Schur Algorithm the coefficients of α_0 ,β_0 are processed in segments, each segment being twice the size of the previous segment.

Schur's Algorithm

Generalized Schur Algorithm



3. Numerical Experiments

We have a working FORTRAN code for the Generalized Schur Algorithm for real data based on the procedure described in [4]. Our code implements the Generalized Schur Algorithm in real arithmetic, and uses dual split-radix FFT algorithms to avoid the explicit use of the bit-reversal permutation. The code uses $6n + O(1)$ real storage locations, and an additional $3n/2$ real storage locations for the Toeplitz matrix and the roots of unity (sines and cosines) required by the FFTs. The nth monic Szegö polynomial χ_n and its squared norm δ_n, where $n = 2^\nu$, are obtained using $(8/3)n\log_2^2 n + O(n\log_2 n)$ real multiplications and $(16/3)n\log_2^2 n + O(n\log_2 n)$ real additions. We plan to make our code available to the scientific community soon.

In [4] it is shown that χ_n, δ_n can be obtained using *less than* $8\,n\,\log_2^2 n$ total real arithmetic operations. To our knowledge, this is the smallest operation count obtained for a superfast Toeplitz solver. Since the Szegö recursions require more than $2\,n^2$ operations, the Generalized Schur Algorithm requires less computational work for $n = 2^\nu \geq 256$. However, our code does not incorporate the considerations discussed in [4, Section 5]. The operation count for our code is therefore somewhat more than the operation count of [4] in the lower order terms.

When the Gaussian or Choleski factorization is used to solve the system of equations $A\,x = b$, the computed solution $\hat{x}$ satisfies $(A+E)\,\hat{x} = b$ with $\|E\| \leq \epsilon\,\phi(n)\,\|A\|$, where ϵ is the machine precision and $\phi(n)$ is a slowly growing function of n. It follows that the residual $r = b - A\,\hat{x} = E\,\hat{x}$ and the error $e = A^{-1}E\,\hat{x}$ satisfy

$$\|r\| \leq \epsilon\,\phi(n)\,\|A\|\,\|\hat{x}\|, \qquad \|e\| \leq \epsilon\,\phi(n)\,\|A\|\,\|A^{-1}\|\,\|\hat{x}\|.$$

Cybenko [9, 10] has shown that $\phi(n)$ is of order n^2 for the Szegö recursions (i.e., for the Levinson-Durbin algorithm for the Yule-Walker equations). As the Choleski decomposition method also yields $\phi(n)$ of order n^2, we can conclude that the accuracy of the Szegö recursions is comparable with that of Choleski's

method. Accurate results can therefore be expected from the Szegö recursions when the positive definite Toeplitz matrix is well conditioned.

We now present some numerical experiments to compare the performance of the Generalized Schur Algorithm with the Szegö recursions. We are pleased to report that the results we have obtained are so far favorable. In particular, the Generalized Schur Algorithm displays roughly the same stability characteristics (e.g., the rates of growth in computed residuals and errors) as the Szegö recursions. The experiments were performed on the VAX 11/750 at Northern Illinois University.

We performed our experiments using the leading principal submatrices of two classes of real positive definite Toeplitz matrices $M_{n+1} = [\mu_{j-k}]_{j,k=0}^{n} = M_{n+1}^T$. We refer to these classes as type A and type B.

$A(n,\theta)$ refers to M_{n+1} where $\mu_j = \theta^{j^2}$, $-1<\theta<1$. These matrices have a particularly special structure [14]. The Schur parameters are given by $\gamma_j = (-\theta)^j$. For fixed n these matrices become ill-conditioned as θ tends to ± 1. For fixed θ the condition numbers tend toward a limiting value as n increases. These are numerically banded matrices, where the bandwidth depends on θ and the smallest positive machine number.

$B(n,\theta_1,\theta_2,\rho)$ refers to the autocorrelation matrix M_{n+1} of order $n+1$ of a vector $x = [\xi_j]_0^{16n-1}$ given by

$$\xi_j = \eta_j + \sum_{k=1}^{n/4} \cos(j\,\omega_k),$$

where th ω_k are $n/4$ equispaced integral multiples of $\pi/8n$ in $[\theta_1, \theta_2]$ $(0 \le \theta_1 < \theta_2 \le \pi/2)$, and η_j is randomly generated in $[-\rho,\rho]$. In other words, $M_{n+1} = T^T\,T$, where T is the Toeplitz matrix formed by the first $n+1$ columns of the circulant matrix whose first column is x. These matrices become ill-conditioned as ρ tends to zero, and the ill-conditioning accelerates as $\theta_2 - \theta_1$ decreases.

We first compare the accuracy of the Szegö recursions (Sz) and the Generalized Schur Algorithm (GS) for computing the monic Szegö polynomial χ_n and the Schur parameters $[\gamma_j]_1^n$. In Tables 1 and 2 we give the square root of the sum of the squared errors (the 2-norm of the error) as well as the maximum absolute error in the coefficients of χ_n and $[\gamma_j]_1^n$ for the indicated matrices of type A and type B. The results of the Szegö recursions in double-precision arithmetic are used as exact answers for the error calculation. Note that the errors for the two algorithms seem to grow at roughly the same rate as n increases. In well-conditioned problems, the Generalized Schur Algorithm typically results in a somewhat larger error than the Szegö recursions, but usually this difference is less than an order of magnitude. On the other hand, the Generalized Schur Algorithm occasionally yielded more accurate results in the ill-conditioned problems of type B.

TABLE 1
Phase One Errors

A(4096,0.6)

n	Szego polynomial errors				Schur parameter errors			
	2-norm		max		2-norm		max	
	GS	Sz	GS	Sz	GS	Sz	GS	Sz
4	0.184e-06	0.219e-06	0.119e-06	0.149e-06	0.802e-07	0.112e-06	0.596e-07	0.894e-07
8	0.315e-06	0.635e-06	0.209e-06	0.328e-06	0.136e-06	0.213e-06	0.782e-07	0.104e-06
16	0.353e 06	0.738e-06	0.201e-06	0.328e-06	0.181e-06	0.227e-06	0.782e-07	0.104e-06
32	0.557e-06	0.780e-06	0.209e-06	0.343e-06	0.229e-06	0.227e-06	0.785e-07	0.104e-06
64	0.719e-06	0.780e-06	0.179e-06	0.343e-06	0.297e-06	0.227e-06	0.988e-07	0.104e-06
128	0.983e-06	0.780e-06	0.184e-06	0.343e-06	0.371e-06	0.227e-06	0.988e-07	0.104e-06
256	0.120e-05	0.780e-06	0.238e-06	0.343e-06	0.436e-06	0.227e-06	0.988e-07	0.104e-06
512	0.149e-05	0.780e-06	0.358e-06	0.343e-06	0.512e-06	0.227e-06	0.988e-07	0.104e-06
1024	0.175e-05	0.780e-06	0.477e-06	0.343e-06	0.579e-06	0.227e-06	0.988e-07	0.104e-06
2048	0.199e-05	0.780e-06	0.596e-06	0.343e-06	0.639e-06	0.227e-06	0.988e-07	0.104e-06
4096	0.225e-05	0.780e-06	0.775e-06	0.343e-06	0.698e-06	0.227e-06	0.988e-07	0.104e-06

A(4096,0.75)

n	Szego polynomial errors				Schur parameters errors			
	2-norm		max		2-norm		max	
	GS	Sz	GS	Sz	GS	Sz	GS	Sz
4	0.333e-06	0.720e-06	0.238e-06	0.536e-06	0.942e-07	0.321e-06	0.894e-07	0.298e-06
8	0.300e-05	0.112e-04	0.155e-05	0.572e-05	0.780e-06	0.168e-05	0.745e-06	0.991e-06
16	0.475e-04	0.276e-04	0.187e-04	0.111e-04	0.473e-05	0.249e-05	0.205e-05	0.991e-06
32	0.521e-04	0.292e-04	0.197e-04	0.113e-04	0.502e-05	0.252e-05	0.205e-05	0.991e-06
64	0.591e-04	0.293e-04	0.197e-04	0.113e-04	0.614e-05	0.252e-05	0.205e-05	0.991e-06
128	0.696e-04	0.293e-04	0.197e-04	0.113e-04	0.736e-05	0.252e-05	0.205e-05	0.991e-06
256	0.835e-04	0.293e-04	0.197e-04	0.113e-04	0.866e-05	0.252e-05	0.205e-05	0.991e-06
512	0.986e-04	0.293e-04	0.198e-04	0.113e-04	0.100e-04	0.252e-05	0.205e-05	0.991e-06
1024	0.120e-03	0.293e-04	0.198e-04	0.113e-04	0.118e-04	0.252e-05	0.205e-05	0.991e-06
2048	0.140e-03	0.293e-04	0.198e-04	0.113e-04	0.135e-04	0.252e-05	0.205e-05	0.991e-06
4096	0.155e-03	0.293e-04	0.199e-04	0.113e-04	0.148e-04	0.252e-05	0.205e-05	0.991e-06

A(4096,0.83)

n	Szego polynomial errors				Schur parameters errors			
	2-norm		max		2-norm		max	
	GS	Sz	GS	Sz	GS	Sz	GS	Sz
4	0.449e-05	0.190e-05	0.286e-05	0.143e-05	0.136e-05	0.786e-06	0.131e-05	0.775e-06
8	0.369e-04	0.173e-03	0.224e-04	0.918e-04	0.420e-05	0.147e-04	0.229e-05	0.115e-04
16	0.180e-02	0.290e-02	0.785e-03	0.114e-02	0.876e-04	0.775e-04	0.643e-04	0.321e-04
32	0.231e-01	0.636e-02	0.598e-02	0.189e-02	0.474e-03	0.112e-03	0.157e-03	0.321e-04
64	0.309e-01	0.655e-02	0.678e-02	0.190e-02	0.638e-03	0.113e-03	0.159e-03	0.321e-04
128	0.444e-01	0.655e-02	0.678e-02	0.190e-02	0.894e-03	0.113e-03	0.159e-03	0.321e-04
256	0.547e-01	0.655e-02	0.679e-02	0.190e-02	0.109e-02	0.113e-03	0.159e-03	0.321e-04
512	0.629e-01	0.655e-02	0.680e-02	0.190e-02	0.122e-02	0.113e-03	0.159e-03	0.321e-04
1024	0.682e-01	0.655e-02	0.681e-02	0.190e-02	0.131e-02	0.113e-03	0.159e-03	0.321e-04
2048	0.725e-01	0.655e-02	0.681e-02	0.190e-02	0.140e-02	0.113e-03	0.159e-03	0.321e-04
4096	0.769e-01	0.655e-02	0.682e-02	0.190e-02	0.148e-02	0.113e-03	0.159e-03	0.321e-04

TABLE 2
Phase One Errors

$B(4096, 20^\circ, 160^\circ, 10.0)$

	Szego polynomial errors				Schur parameters errors			
n	2-norm		max		2-norm		max	
	GS	Sz	GS	Sz	GS	Sz	GS	Sz
4	0.507e-08	0.314e-07	0.301e-08	0.310e-07	0.999e-08	0.999e-08	0.952e-08	0.952e-08
8	0.380e-07	0.149e-06	0.659e-07	0.838e-07	0.357e-07	0.933e-07	0.242e-07	0.838e-07
16	0.422e-06	0.850e-06	0.185e-06	0.411e-06	0.134e-06	0.287e-06	0.888e-07	0.158e-06
32	0.776e-06	0.878e-06	0.294e-06	0.404e-06	0.208e-06	0.340e-06	0.888e-07	0.158e-06
64	0.186e-05	0.918e-06	0.532e-06	0.387e-06	0.499e-06	0.413e-06	0.151e-06	0.158e-06
128	0.256e-05	0.111e-05	0.841e-06	0.504e-06	0.587e-06	0.434e-06	0.151e-06	0.158e-06
256	0.399e-05	0.138e-05	0.138e-05	0.692e-06	0.683e-06	0.477e-06	0.151e-06	0.158e-06
512	0.670e-05	0.181e-05	0.238e-05	0.734e-06	0.781e-06	0.540e-06	0.151e-06	0.158e-06
1024	0.115e-04	0.246e-05	0.441e-05	0.955e-06	0.104e-05	0.656e-06	0.151e-06	0.158e-06
2048	0.205e-04	0.344e-05	0.811e-05	0.966e-06	0.139e-05	0.919e-06	0.151e-06	0.158e-06
4096	0.854e-04	0.270e-04	0.297e-04	0.119e-04	0.547e-04	0.168e-04	0.250e-04	0.616e-05

$B(4096, 20^\circ, 160^\circ, 0.5)$

	Szego polynomial errors				Schur parameters errors			
n	2-norm		max		2-norm		max	
	GS	Sz	GS	Sz	GS	Sz	GS	Sz
4	0.346e-07	0.587e-07	0.257e-07	0.515e-07	0.328e-07	0.570e-07	0.239e-07	0.515e-07
8	0.945e-07	0.474e-06	0.596e-07	0.232e-06	0.625e-07	0.261e-06	0.527e-07	0.232e-06
16	0.807e-05	0.377e-04	0.284e-05	0.144e-04	0.116e-05	0.527e-05	0.873e-06	0.430e-05
32	0.111e-02	0.480e-02	0.406e-03	0.195e-02	0.467e-04	0.205e-03	0.204e-04	0.937e-04
64	0.436e-02	0.499e-02	0.172e-02	0.203e-02	0.148e-03	0.312e-03	0.588e-04	0.999e-04
128	0.501e-02	0.524e-02	0.156e-02	0.187e-02	0.185e-03	0.362e-03	0.588e-04	0.999e-04
256	0.666e-02	0.848e-02	0.153e-02	0.163e-02	0.238e-03	0.490e-03	0.588e-04	0.999e-04
512	0.796e-02	0.182e-01	0.158e-02	0.170e-02	0.280e-03	0.809e-03	0.588e-04	0.999e-04
1024	0.139e-01	0.316e-01	0.189e-02	0.231e-02	0.428e-03	0.140e-02	0.588e-04	0.999e-04
2048	0.461e-01	0.659e-01	0.303e-02	0.353e-02	0.145e-02	0.311e-02	0.116e-03	0.173e-03
4096	0.506e-01	0.413e+00	0.131e-01	0.104e+00	0.376e-01	0.221e+00	0.132e-01	0.830e-01

$B(4096, 50^\circ, 100^\circ, 1.0)$

	Szego polynomial errors				Schur parameters errors			
n	2-norm		max		2-norm		max	
	GS	Sz	GS	Sz	GS	Sz	GS	Sz
4	0.393e-05	0.409e-05	0.238e-05	0.262e-05	0.169e-05	0.177e-05	0.161e-05	0.173e-05
8	0.490e-04	0.393e-04	0.312e-04	0.277e-04	0.270e-04	0.255e-04	0.206e-04	0.187e-04
16	0.124e-03	0.307e-03	0.830e-04	0.154e-03	0.585e-04	0.120e-03	0.427e-04	0.631e-04
32	0.319e-03	0.500e-03	0.140e-03	0.170e-03	0.129e-03	0.224e-03	0.480e-04	0.757e-04
64	0.411e-03	0.647e-03	0.120e-03	0.165e-03	0.177e-03	0.333e-03	0.480e-04	0.757e-04
128	0.644e-03	0.890e-03	0.141e-03	0.197e-03	0.245e-03	0.479e-03	0.634e-04	0.757e-04
256	0.952e-03	0.129e-02	0.165e-03	0.246e-03	0.346e-03	0.676e-03	0.634e-04	0.783e-04
512	0.152e-02	0.231e-02	0.196e-03	0.309e-03	0.530e-03	0.104e-02	0.634e-04	0.989e-04
1024	0.260e-02	0.325e-02	0.231e-03	0.357e-03	0.904e-03	0.155e-02	0.726e-04	0.114e-03
2048	0.303e-02	0.439e-02	0.254e-03	0.396e-03	0.122e-02	0.217e-02	0.726e-04	0.114e-03
4096	0.395e-02	0.641e-02	0.282e-03	0.436e-03	0.173e-02	0.312e-02	0.769e-04	0.114e-03

Our second accuracy experiment compares the Szegö recursions and the Generalized Schur Algorithm for the solution of a Toeplitz system of equations. The Gohberg-Semencul formula is used for the solution phase for both algorithms.

Let $\|M\|_1$ and $\|M\|_2$ denote the norm of the matrix M that is induced by the vector 1-norm and vector 2-norm, respectively. Then

$$\max_{1 \leq j \leq n} \|M\ e_j\|_2 \leq \|M\|_2 \leq \sqrt{n}\ \ \|M\|_1. \tag{3.1}$$

Let $M = M_n$, and define $\tilde{R} := R\ D^{-1/2}$ so that $M^{-1} = \tilde{R}\ \tilde{R}^*$. It is shown in [2] that

$$\|\tilde{R} e_n\|_2^2 \leq \|M^{-1}\|_2 \leq n\ \ \|\tilde{R}\ e_n\|_2^2\ . \tag{3.2}$$

This gives bounds on the 2-norm of M^{-1} in terms of the last column of $\tilde{R}$ (i.e., in terms of the coefficients of the normalized Szegö polynomial $\chi_{n-1}/\sqrt{\delta_{n-1}}$).

After computing a solution $\hat{x}$ of the positive definite Toeplitz system, the residual $r = b - M\ \hat{x}$ is calculated in double precision, and the system $M\ e = r$ is solved (in $O(n \lg n)$ operations) to obtain an estimate $\hat{e}$ of the error. Finally, the upper and lower bounds for the two ratios $\psi_r = \dfrac{\|r\|}{\|M\|\ \|\hat{x}\|}$ and $\psi_e = \dfrac{\|\hat{e}\|}{\|M\|\ \|M^{-1}\|\ \|\hat{x}\|}$ are obtained using (3.1) and (3.2). In order for an algorithm to be stable, these ratios should increase slowly with n. In Tables 3 and 4 we display the computed lower bounds on $\|M^{-1}\|_2$, the computed relative residuals and relative errors, and upper bounds on the ratios ψ_r and ψ_e. These results were obtained using $b = [j+1]_{j=0}^{n}$. Similar results were obtained when the components of b were set equal to one and when they were randomly generated. The Szegö recursions were stopped at $n = 2^{13}$.

Note that the sizes of the residuals and errors for each algorithm are roughly the same, as are the rates of growth in these quantities. The upper bounds on ψ_r and ψ_e are slowly growing for both algorithms. The errors and residuals for the

TABLE 3
Toeplitz Solver Residuals and Errors

$A(2^{15}, 0.6)$

n	$\|M_{n+1}^{-1}\|_2$	$\frac{\|r\|}{\|b\|}$ GS	$\frac{\|r\|}{\|b\|}$ Sz	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ GS	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ Sz	ψ_r GS	ψ_r Sz	ψ_e GS	ψ_e Sz
4	0.449e+01	0.376e-06	0.580e-06	0.353e-06	0.483e-06	0.376e-06	0.580e-06	0.594e-07	0.812e-07
8	0.493e+01	0.221e-06	0.615e-06	0.253e-06	0.631e-06	0.266e-06	0.740e-06	0.388e-07	0.967e-07
16	0.495e+01	0.383e-06	0.444e-06	0.594e-06	0.898e-06	0.545e-06	0.631e-06	0.906e-07	0.137e-06
32	0.495e+01	0.280e-06	0.457e-06	0.540e-06	0.677e-06	0.449e-06	0.731e-06	0.825e-07	0.103e-06
64	0.495e+01	0.725e-06	0.117e-06	0.910e-06	0.711e-06	0.125e-05	0.202e-06	0.139e-06	0.109e-05
128	0.495e+01	0.567e-06	0.296e-06	0.120e-05	0.107e-05	0.102e-05	0.529e-06	0.184e-06	0.163e-06
256	0.495e+01	0.105e-05	0.205e-06	0.171e-05	0.140e-05	0.192e-05	0.375e-06	0.261e-06	0.214e-06
512	0.495e+01	0.150e-05	0.327e-06	0.161e-05	0.697e-06	0.278e-05	0.604e-06	0.246e-06	0.106e-06
1024	0.495e+01	0.206e-05	0.399e-06	0.222e-05	0.893e-06	0.383e-05	0.743e-06	0.338e-06	0.136e-06
2048	0.495e+01	0.288e-05	0.446e-06	0.299e-05	0.934e-06	0.538e-05	0.833e-06	0.457e-06	0.143e-06
4096	0.495e+01	0.410e-05	0.501e-06	0.418e-05	0.944e-06	0.767e-05	0.937e-06	0.638e-06	0.144e-06
8192	0.495e+01	0.595e-05	0.683e-06	0.600e-05	0.109e-05	0.111e-04	0.128e-05	0.916e-06	0.167e-06
16384	0.495e+01	0.571e-05		0.590e-05		0.107e-04		0.901e-06	
32768	0.495e+01	0.723e-05		0.738e-05		0.135e-04		0.113e-05	

$A(2^{15}, 0.75)$

n	$\|M_{n+1}^{-1}\|_2$	$\frac{\|r\|}{\|b\|}$ GS	$\frac{\|r\|}{\|b\|}$ Sz	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ GS	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ Sz	ψ_r GS	ψ_r Sz	ψ_e GS	ψ_e Sz
4	0.279e+02	0.703e-06	0.122e-05	0.434e-06	0.169e-05	0.321e-06	0.557e-06	0.102e-07	0.397e-07
8	0.647e+02	0.704e-06	0.591e-06	0.278e-05	0.514e-05	0.316e-06	0.265e-06	0.281e-07	0.520e-07
16	0.799e+02	0.170e-05	0.163e-05	0.905e-05	0.115e-04	0.967e-06	0.923e-06	0.741e-07	0.942e-07
32	0.803e+02	0.123e-05	0.425e-05	0.140e-04	0.117e-04	0.959e-06	0.331e-05	0.114e-06	0.954e-07
64	0.803e+02	0.386e-05	0.285e-05	0.161e-04	0.142e-04	0.402e-05	0.296e-05	0.131e-06	0.116e-06
128	0.803e+02	0.513e-05	0.293e-05	0.289e-04	0.260e-04	0.681e-05	0.389e-05	0.235e-06	0.212e-06
256	0.803e+02	0.922e-05	0.255e-05	0.520e-04	0.504e-04	0.148e-04	0.408e-05	0.424e-06	0.411e-06
512	0.803e+02	0.384e-05	0.286e-05	0.267e-04	0.207e-04	0.699e-05	0.521e-05	0.217e-06	0.169e-06
1024	0.803e+02	0.610e-05	0.300e-05	0.372e-04	0.315e-04	0.120e-04	0.590e-05	0.303e-06	0.257e-06
2048	0.803e+02	0.891e-05	0.288e-05	0.363e-04	0.355e-04	0.183e-04	0.593e-05	0.296e-06	0.289e-06
4096	0.803e+02	0.951e-05	0.322e-05	0.317e-04	0.303e-04	0.200e-04	0.679e-05	0.258e-06	0.246e-06
8192	0.803e+02	0.692e-05	0.334e-05	0.338e-04	0.341e-04	0.148e-04	0.714e-05	0.275e-06	0.278e-06
16384	0.803e+02	0.107e-04		0.744e-04		0.231e-04		0.606e-06	
32768	0.803e+02	0.159e-04		0.738e-04		0.344e-04		0.601e-06	

$A(2^{15}, 0.8)$

n	$\|M_{n+1}^{-1}\|_2$	$\frac{\|r\|}{\|b\|}$ GS	$\frac{\|r\|}{\|b\|}$ Sz	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ GS	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ Sz	ψ_r GS	ψ_r Sz	ψ_e GS	ψ_e Sz
4	0.721e+02	0.161e-05	0.265e-05	0.741e-06	0.129e-05	0.448e-06	0.739e-06	0.636e-08	0.110e-07
8	0.320e+03	0.461e-05	0.546e-05	0.661e-05	0.213e-05	0.932e-06	0.110e-05	0.127e-07	0.408e-08
16	0.610e+03	0.462e-05	0.124e-04	0.327e-04	0.826e-05	0.985e-06	0.265e-05	0.329e-07	0.831e-08
32	0.642e+03	0.764e-05	0.213e-04	0.137e-03	0.190e-04	0.225e-05	0.628e-05	0.131e-06	0.182e-07
64	0.642e+03	0.112e-04	0.192e-04	0.162e-03	0.400e-04	0.468e-05	0.800e-05	0.155e-06	0.383e-07
128	0.642e+03	0.427e-05	0.154e-04	0.190e-03	0.123e-03	0.249e-05	0.900e-05	0.181e-06	0.118e-06
256	0.642e+03	0.126e-04	0.172e-04	0.279e-03	0.291e-03	0.101e-04	0.138e-04	0.267e-06	0.278e-06
512	0.642e+03	0.113e-04	0.174e-04	0.264e-03	0.129e-03	0.121e-04	0.186e-04	0.253e-06	0.123e-06
1024	0.642e+03	0.876e-05	0.172e-04	0.363e-03	0.264e-03	0.120e-04	0.237e-04	0.347e-06	0.253e-06
2048	0.642e+03	0.486e-05	0.175e-04	0.392e-03	0.301e-03	0.812e-05	0.292e-04	0.375e-06	0.288e-06
4096	0.642e+03	0.817e-05	0.150e-04	0.367e-03	0.295e-03	0.156e-04	0.288e-04	0.351e-06	0.282e-06
8192	0.642e+03	0.116e-04	0.178e-04	0.421e-03	0.372e-03	0.241e-04	0.371e-04	0.403e-06	0.356e-06
16384	0.642e+03	0.221e-04		0.891e-03		0.482e-04		0.852e-06	
32768	0.642e+03	0.204e-04		0.906e-03		0.458e-04		0.867e-06	

TABLE 4
Toeplitz Solver Residuals and Errors

$B(2^{15}, 20^\circ, 160^\circ, 50)$

n	$\|M_{n+1}^{-1}\|_2$	$\frac{\|r\|}{\|b\|}$ GS	$\frac{\|r\|}{\|b\|}$ Sz	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ GS	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ Sz	ψ_r GS	ψ_r Sz	ψ_e GS	ψ_e Sz
4	0.133e+01	0.824e-07	0.824e-07	0.760e-07	0.760e-07	0.545e-07	0.545e-07	0.538e-07	0.538e-07
8	0.178e+01	0.211e-06	0.235e-06	0.136e-06	0.249e-06	0.828e-07	0.922e-07	0.701e-07	0.129e-06
16	0.192e+01	0.372e-06	0.221e-06	0.112e-06	0.170e-06	0.763e-07	0.453e-07	0.529e-07	0.806e-07
32	0.200e+01	0.515e-06	0.305e-06	0.468e-06	0.173e-06	0.889e-07	0.526e-07	0.212e-06	[illegible]
64	0.203e+01	0.408e-06	0.248e-06	0.198e-06	0.878e-07	0.657e-07	0.399e-07	0.883e-07	[illegible]
128	0.205e+01	0.538e-06	0.389e-06	0.270e-06	0.224e-06	0.840e-07	0.607e-07	0.119e-06	0.988e-07
256	0.206e+01	0.101e-05	0.736e-06	0.904e-06	0.650e-06	0.156e-06	0.114e-06	0.396e-06	0.285e-06
512	0.206e+01	0.395e-05	0.771e-06	0.393e-05	0.677e-06	0.612e-06	0.120e-06	0.172e-05	0.296e-06
1024	0.207e+01	0.115e-04	0.117e-05	0.115e-04	0.110e-05	0.175e-05	0.177e-06	0.501e-05	0.482e-06
2048	0.208e+01	0.228e-04	0.514e-05	0.229e-04	0.514e-05	0.339e-05	0.763e-06	0.992e-05	0.223e-05
4096	0.209e+01	0.501e-04	0.164e-04	0.501e-04	0.164e-04	0.723e-05	0.237e-05	0.216e-04	0.708e-05
8192	0.212e+01	0.102e-03	0.375e-04	0.102e-03	0.375e-04	0.144e-04	0.532e-05	0.432e-04	0.160e-04
16384	0.217e+01	0.196e-03		0.197e-03	0.000e+00	0.265e-04	0.000e+00	0.814e-04	0.000e+00
32768	0.614e+01	0.469e-03		0.470e-03	0.000e+00	0.399e-04	0.000e+00	0.520e-04	0.000e+00

$B(2^{15}, 20^\circ, 160^\circ, 5)$

n	$\|M_{n+1}^{-1}\|_2$	$\frac{\|r\|}{\|b\|}$ GS	$\frac{\|r\|}{\|b\|}$ Sz	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ GS	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ Sz	ψ_r GS	ψ_r Sz	ψ_e GS	ψ_e Sz
4	0.158e+01	0.847e-07	0.942e-07	0.964e-07	0.119e-06	0.484e-07	0.539e-07	0.560e-07	0.689e-07
8	0.382e+01	0.304e-06	0.262e-06	0.192e-06	0.171e-06	0.732e-07	0.631e-07	0.448e-07	0.398e-07
16	0.440e+02	0.486e-05	0.314e-05	0.338e-05	0.559e-06	0.137e-06	0.886e-07	0.676e-07	0.112e-07
32	0.974e+02	0.273e-04	0.272e-04	0.201e-04	0.112e-04	0.696e-07	0.693e-07	0.180e-06	0.100e-06
64	0.112e+03	0.391e-04	0.285e-04	0.266e-04	0.161e-04	0.810e-07	0.590e-07	0.207e-06	0.125e-06
128	0.118e+03	0.423e-04	0.363e-04	0.299e-04	0.194e-04	0.800e-07	0.687e-07	0.220e-06	0.143e-06
256	0.121e+03	0.600e-04	0.426e-04	0.471e-04	0.201e-04	0.109e-06	0.777e-07	0.337e-06	0.144e-06
512	0.123e+03	0.729e-04	0.457e-04	0.662e-04	0.196e-04	0.132e-06	0.827e-07	0.468e-06	0.138e-06
1024	0.124e+03	0.914e-04	0.484e-04	0.830e-04	0.189e-04	0.161e-06	0.855e-07	0.582e-06	0.133e-06
2048	0.124e+03	0.933e-04	0.456e-04	0.833e-04	0.144e-04	0.160e-06	0.783e-07	0.580e-06	0.101e-06
4096	0.126e+03	0.675e-04	0.472e-04	0.500e-04	0.705e-05	0.113e-06	0.787e-07	0.345e-06	0.487e-07
8192	0.128e+03	0.607e-04	0.753e-04	0.235e-04	0.530e-04	0.996e-07	0.124e-06	0.160e-06	0.360e-06
16384	0.132e+03	0.924e-04		0.703e-04	0.000e+00	0.144e-06	0.000e+00	0.462e-06	0.000e+00
32768	0.535e+03	0.922e-03		0.129e-03	0.000e+00	0.850e-06	0.000e+00	0.148e-06	0.000e+00

$B(2^{15}, 70^\circ, 110^\circ, 1.0)$

n	$\|M_{n+1}^{-1}\|_2$	$\frac{\|r\|}{\|b\|}$ GS	$\frac{\|r\|}{\|b\|}$ Sz	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ GS	$\frac{\|\hat{e}\|}{\|\hat{x}\|}$ Sz	ψ_r GS	ψ_r Sz	ψ_e GS	ψ_e Sz
4	0.276e+03	0.223e-04	0.188e-04	0.513e-05	0.109e-04	0.717e-07	0.606e-07	0.113e-07	0.239e-07
8	0.771e+04	0.654e-03	0.147e-02	0.273e-03	0.125e-03	0.482e-07	0.109e-06	0.183e-07	0.840e-08
16	0.819e+04	0.720e-03	0.148e-02	0.332e-03	0.238e-03	0.342e-07	0.704e-07	0.200e-07	0.143e-07
32	0.872e+04	0.118e-02	0.161e-02	0.757e-03	0.662e-03	0.505e-07	0.687e-07	0.417e-07	0.364e-07
64	0.926e+04	0.222e-02	0.179e-02	0.141e-02	0.986e-03	0.905e-07	0.728e-07	0.722e-07	0.504e-07
128	0.939e+04	0.331e-02	0.208e-02	0.270e-02	0.124e-02	0.132e-06	0.831e-07	0.136e-06	0.619e-07
256	0.953e+04	0.419e-02	0.209e-02	0.394e-02	0.122e-02	0.165e-06	0.821e-07	0.194e-06	0.600e-07
512	0.959e+04	0.584e-02	0.196e-02	0.554e-02	0.994e-03	0.226e-06	0.755e-07	0.271e-06	0.483e-07
1024	0.963e+04	0.669e-02	0.201e-02	0.645e-02	0.888e-03	0.254e-06	0.759e-07	0.313e-06	0.430e-07
2048	0.969e+04	0.749e-02	0.202e-02	0.720e-02	0.826e-03	0.283e-06	0.759e-07	0.348e-06	0.398e-07
4096	0.978e+04	0.825e-02	0.194e-02	0.785e-02	0.739e-03	0.318e-06	0.743e-07	0.375e-06	0.352e-07
8192	0.101e+05	0.831e-02	0.178e-02	0.117e-01	0.715e-03	0.323e-06	0.684e-07	0.544e-06	0.336e-07
16384	0.103e+05	0.302e-02		0.789e-02		0.333e-06		0.358e-06	
32768	0.116e+05	0.804e-02		0.277e-01		0.400e-06		0.111e-05	

Generalized Schur Algorithm were typically larger than those of Szegö recursions in the well-conditioned problems, while the Generalized Schur Algorithm sometimes performed slightly better in the relatively ill-conditioned problems. These differences were rarely more than an order of magnitude.

Thus, in the examples we investigated the accuracy of the Generalized Schur Algorithm closely followed that of Szegö recursions. Of course, while we did not encounter any problems where the accuracy of the two algorithms was greatly different, more experimentation is needed to obtain a better confidence in the results of the Generalized Schur Algorithm.

Table 5 shows average CPU times required by the Generalized Schur Algorithm and Szegö recursions for the calculation of the nth Szegö polynomial and the n Schur parameters. Note that the Szegö recursions would have used about 36 CPU hours for $n = 2^{16}$, compared with about 38 minutes for the Generalized Schur Algorithm. The dramatic relative efficiency of the Generalized Schur Algorithm for large problems and the potential reliability of the algorithm as indicated in the above examples lead us to believe the algorithm will become a popular method for solving huge positive definite Toeplitz systems of equations.

TABLE 5
Timing comparison (CPU seconds)

n	GS	Sz
256	3.8	2.4
512	8.1	8.6
1024	17.8	32.2
2048	40.0	126.0
4096	90.7	505.8
8192	200.8	2030.9
16384	441.2	8068.3
32768	1001.6	32222.3
65536	2257.4	--

References

[1] N. I. Akhiezer, The Classical Moment Problem, Oliver and Boyd, Edinburgh, 1965.

[2] G. S. Ammar and W. B. Gragg. *Implementation and Use of the Generalized Schur Algorithm,* in Computational and Combinatorial Methods in Systems Theory, (C. I. Byrnes and A. Lindquist, editors) North-Holland, Amsterdam, 1986, pp. 265-280.

[3] G. S. Ammar and W. B. Gragg. *The Generalized Schur Algorithm for the Superfast Solution of Toeplitz Systems,* in Rational Approximation and its Applications in Mathematics and Physics (J. Gilewicz, M. Pindor and W. Siemaszko, editors), Lecture Notes in Mathematics 1237, Springer, Berlin, 1987, pp. 315-330.

[4] G. S. Ammar and W. B. Gragg, *Superfast Solution of Real Positive Definite Toeplitz Systems,* SIAM J. Matrix Analysis and Applications, to appear.

[5] E. H. Bareiss, *Numerical Solution of Linear Equations with Toeplitz and vector Toeplitz Matrices,* Numer. Math., 13 (1969), 404-424.

[6] R. R. Bitmead and B. D. O. Anderson, *Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations,* Linear Algebra Appl., 34 (1980), 103-116.

[7] R. P. Brent. F. G. Gustavson and D. Y. Y. Yun, *Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximants,* J. Algorithms, 1 (1980) 259-295.

[8] J. R. Bunch, *Stability of Methods for Solving Toeplitz Systems of Equations,* SIAM J. Sci. Statist. Comput., 6 (1985), 349-364.

[9] G. Cybenko, Error Analysis of some Signal Processing Algorithms, Ph. D. Thesis, Princeton Univ., Princeton, NJ, 1978.

[10] G. Cybenko, *The Numerical Stability of the Levinson-Durbin Algorithm for Toeplitz Systems of Equations,* SIAM J. Sci. Stat. Comput., 1 (1980), 303-319.

[11] J.-M. Delosme and M. Morf, *Normalized Doubling Algorithms for Finite Shift-Rank Processes,* Proceedings of the 20th IEEE Conference on Decision and Control, San Diego, CA, 1981, 346-348.

[12] I. C. Gohberg and I. A. Fel′dman, Convolution Equations and Projection Methods for their Solution, American Mathematical Society, Providence, RI, 1974.

[13] G. H. Golub and C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, MD, 1984.

[14] W. B. Gragg, *Positive definite Toeplitz Matrices, the Arnoldi Process for Isometric Operators, and Gaussian Quadrature on the Unit Circle (in Russian)*, in Numerical Methods in Linear Algebra (E. S. Nikolaev editor), Moscow University Press, 1982, 16-32.

[15] W. B. Gragg, F. G. Gustavson, D. D. Warner and D. Y. Y. Yun, *On Fast Computation of Superdiagonal Padé Fractions*, Math. Programming Stud., 18 (1982), 39-42.

[16] U. Grenander and G. Szegö, Toeplitz Forms and their Applications, Chelsea, New York, 1984.

[17] F. de Hoog, *A New Algorithm for Solving Toeplitz Systems of Equations*, Lin. Algebra Appl., 88/89 (1987), 122-138.

[18] J. R. Jain, *An Efficient Algorithm for a Large Toeplitz Set of Linear Equations*, IEEE Trans. Acoust. Speech Signal Process., 27 (1979), 612-615.

[19] T. Kailath, *A Theorem of I. Schur and Its Impact on Modern Signal Processing*, in I. Schur Methods in Operator Theory and Signal Processing, I. Gohberg, editor, Birkhäuser Verlag, Basel, 1986, pp. 9-30.

[20] T. Kailath, A. Vieira and M. Morf, *Inverses of Toeplitz Operators, Innovations, and Orthogonal Polynomials*, SIAM Rev., 20 (1978), 106-119.

[21] N. Levinson, *The Wiener RMS (Root-Mean-Square) Error Criterion in Filter Design and Prediction*, J. Math. Phys., 25 (1947), 261-278.

[22] B. R. Musicus, *Levinson and Fast Choleski Algorithms for Toeplitz and Almost Toeplitz Matrices*, Report, Res. Lab. of Electronics, M.I.T., 1984.

[23] I. Schur, *Uber Potenzreihen, die in Innern des Einheitskrises Beschränkt Sind*, J. Reine Angew. Math., 147 (1917), 205-232.

[24] W. Trench, *An Algorithm for the Inversion of Finite Toeplitz Matrices*, SIAM J. Appl. Math., 12 (1964), 515-522.

DISTRIBUTION LIST

DIRECTOR DEFENSE TECH. INFORMATION CENTER, CAMERON STATION ALEXANDRIA, VA 22314	(2)
DIRECTOR OF RESEARCH ADMIN. CODE 012 NAVAL POSTGRADUATE SCHOOL MONTEREY, CA 93943	
LIBRARY CODE 0142 NAVAL POSTGRADUATE SCHOOL MONTEREY, CA 93943	(2)
DEPARTMENT OF MATHEMATICS CODE 53 NAVAL POSTGRADUATE SCHOOL MONTEREY, CA 93943	
CENTER FOR NAVAL ANALYSES 4401 FORD AVENUE ALEXANDRIA, VA 22302-0268	
PROFESSOR WILLIAM GRAGG CODE 53Gr DEPARTMENT OF MATHEMATICS NAVAL POSTGRADUATE SCHOOL MONTEREY, CA 93943	(15)
NATIONAL SCIENCE FOUNDATION WASHINGTON, D.C. 20550	